

Oracle Day 10 – SQL Types

Note: Please watch my YouTube sessions to better understand the descriptions and queries below

NiC IT Academy YouTube Videos for reference

● Oracle SQL Tutorial - English

https://youtube.com/playlist?list=PLsphD3EpR7F9mmtY2jBt_O8Q9XmvrhQEF

● Oracle SQL - தமிழில்

https://youtube.com/playlist?list=PLsphD3EpR7F-u4Jjp_3fYgLSsKwPPTEH4

✦ Oracle SQL Day wise Video: ENGLISH

Oracle SQL Day 1 – Introduction to Oracle - <https://youtu.be/hLnKjYGr730>

Oracle SQL Day 2 – SQL Types DDL, DML, DRL, DCL, TCL - <https://youtu.be/XpgjXvnfZec>

Oracle SQL Day 3 – Constraints in Oracle - <https://youtu.be/TmYqeFfHyyc>

Oracle SQL Day 4 – SELECT Statements in Oracle - <https://youtu.be/tYQfBgUCpol>

Oracle SQL Day 5 – Single Row Functions in Oracle - <https://youtu.be/4qJxQuHLC4>

Oracle SQL Day 6 – Joins in Oracle - <https://youtu.be/CkaqluC2afE>

Oracle SQL Day 7 – Aggregate Functions in Oracle - <https://youtu.be/BSiCWzj-py8>

Oracle SQL Day 8 – Sub Queries in Oracle - <https://youtu.be/KtUCyG2cZe4>

Oracle SQL Day 9 – SET Operators in Oracle - <https://youtu.be/B0JbGbWsEIA>

Oracle SQL Day 10 – Analytical Functions in Oracle - <https://youtu.be/gRC3ndWLsoo>

Oracle SQL Day 11 - Views in Oracle - <https://youtu.be/m8a1UtOmd5k>

Oracle SQL Day 12 - Indexes in Oracle - <https://youtu.be/reL2O-kvNxc>

Oracle SQL Day 13 - Regular Expression - https://youtu.be/k_Eo08vLPhU



Analytical Functions or Window Functions in Oracle:

=====

1. Rank() -- RANK() over(order by ranking_column asc|desc)

--RANK() over(partition by group_column order by ranking_column asc|desc)

2. Dense_Rank() -- Dense_RANK() over(order by ranking_column asc|desc)

3. Row_number()

4. Lead()

5. Lag()

| SALARY | RANK() | DENSE_RANK() | Row_number |
|--------|--------|--------------|------------|
| 70000 | 1 | 1 | 1 |
| 60000 | 2 | 2 | 2 |
| 50000 | 3 | 3 | 3 |
| 45000 | 4 | 4 | 4 |
| 40000 | 5 | 5 | 5 |
| 40000 | 5 | 5 | 6 |
| 38000 | 7 | 6 | 7 |
| 35000 | 8 | 7 | 8 |
| 35000 | 8 | 7 | 9 |
| 34000 | 10 | 8 | 10 |
| 32000 | 11 | 9 | 11 |
| 20000 | 12 | 10 | 12 |
| 20000 | 12 | 10 | 13 |
| 20000 | 12 | 10 | 14 |
| 18000 | 15 | 11 | 15 |
| 15000 | 16 | 12 | 16 |
| 12000 | 17 | 13 | 17 |



```
=====
```



```
select * from employees;
```

```
select sum(salary) from employees;
```

```
-- 749716
```

```
select employee_id,first_name,salary,department_id,  
sum(salary) over(partition by department_id) ,sum(salary) over() from employees;
```

```
=====
```



```
RANK() OVER(Order by salary desc)
```

```
RANK() OVER(Order by salary desc)
```

```
select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(order by salary) RANK from employees;
```

```
select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(order by salary desc) RANK from employees;
```

```
--dense rank()
```

```
select employee_id,first_name,email,phone_number,salary,department_id,  
dense_rank() over(order by salary) RANK from employees;
```



```
select employee_id,first_name,email,phone_number,salary,department_id,
dense_rank() over(order by salary desc) RANK from employees;
```

```
--row_number()
```

```
select employee_id,first_name,email,phone_number,hire_date,salary,department_id,
row_number() over(order by salary desc) no_ties from employees;
```

```
select employee_id,first_name,email,phone_number,hire_date,salary,department_id,
row_number() over(order by salary desc,hire_date) no_ties from employees;
```

| SALARY | Dept_id | RANK() |
|--------|---------|--------|
| 40000 | 10 | 1 |
| 35000 | 10 | 2 |
| 20000 | 10 | 3 |
| 12000 | 10 | 4 |
| 50000 | 20 | 1 |
| 45000 | 20 | 2 |
| 35000 | 20 | 3 |
| 20000 | 20 | 4 |
| 20000 | 20 | 4 |
| 40000 | 30 | 1 |
| 38000 | 30 | 2 |
| 34000 | 30 | 3 |
| 32000 | 30 | 4 |
| 70000 | 40 | 1 |
| 60000 | 40 | 2 |
| 18000 | 40 | 3 |
| 15000 | 40 | 4 |



group by == > Partition by

```
select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(partition by department_id order by salary) RANK from employees;
```

```
select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(partition by department_id order by salary desc) RANK from employees;
```

```
select employee_id,first_name,email,phone_number,salary,department_id,  
dense_rank() over(partition by department_id order by salary) RANK from employees;
```

```
select employee_id,first_name,email,phone_number,salary,department_id,  
dense_rank() over(partition by department_id order by salary desc) RANK from employees;
```

--Least 5 salaried employees

```
select employee_id,first_name,email,phone_number,salary,department_id,rank from  
(select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(order by salary) RANK from employees) where rank <=5;
```

-- Top 5 earners

```
select employee_id,first_name,email,phone_number,salary,department_id,rank from  
(select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(order by salary desc) RANK from employees) where rank <=5;
```



--Least 5 salaried employees

```
select employee_id,first_name,email,phone_number,salary,department_id,rank from  
(select employee_id,first_name,email,phone_number,salary,department_id,  
dense_rank() over(order by salary) RANK from employees) where rank <=5;
```

-- Top 5 earners

```
select employee_id,first_name,email,phone_number,salary,department_id,rank from  
(select employee_id,first_name,email,phone_number,salary,department_id,  
dense_rank() over(order by salary desc) RANK from employees) where rank <=5;
```

-- Top 5 earners with department_name

```
select a.employee_id,  
a.first_name,  
a.email,  
a.phone_number,  
a.salary,  
a.department_id,  
d.department_name,  
a.rank from  
(select employee_id,first_name,email,phone_number,salary,department_id,  
rank() over(order by salary desc) "RANK" from employees) a inner join departments d  
on a.department_id=d.department_id where a.rank <=5 order by a.rank;
```



100

middle record

select * from employees where rownum <=5;

107

select count(rownum)=count(rownum)/2;

select rownum,e.* from employees e where rownum <=(select round(count(*)/2) from employees)
minus

select rownum,e.* from employees e where rownum <(select round(count(*)/2) from employees);

select * from employees;

Lead()

Lag()

select employee_id,first_name,email,phone_number,hire_date,salary,department_id,
lead (hire_date) over(order by hire_date) after_hire,
lead (first_name) over(order by hire_date) after_hire from employees;



```
select * from (  
select employee_id,first_name,email,phone_number,hire_date,salary,department_id,  
lag (hire_date) over(order by hire_date) before_hire_date,  
lag (first_name) over(order by hire_date) before_hire_name from employees) where employee_id=142;
```

```
select employee_id,first_name,email,phone_number,hire_date,salary,department_id,  
lead (FIRST_NAME) over(order by salary) high_salary_person,  
lead (salary) over(order by salary) high_salary from employees;
```

```
select employee_id,first_name,salary,department_id,sum(salary) over() from employees;
```

```
select employee_id,first_name,salary,department_id,sum(salary) over(partition by department_id),  
sum(salary) over() from employees;
```

-- Non-ANSI Syntax

```
with dept_count as (  
select deptno, count(*) as dept_count  
from emp  
group by deptno)  
select e.ename as employee_name,  
dc.dept_count as emp_dept_count
```




```
from emp e,  
    dept_count dc  
where e.deptno = dc.deptno;
```

-- ANSI Syntax

```
with dept_count as (  
    select deptno, count(*) as dept_count  
    from emp  
    group by deptno)  
select e.ename as employee_name,  
    dc.dept_count as emp_dept_count  
from emp e  
    join dept_count dc on e.deptno = dc.deptno;
```

NiC IT Academy

